CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION CFSTI
DOCUMENT MANAGEMENT BRANCH 410.11

LIMITATIONS IN REPRODUCTION QUALITY

ACCESSION # AD 606935

- [x] 1. WE REGRET THAT LEGIBILITY OF THIS DOCUMENT IS IN PART UNSATISFACTORY. REPRODUCTION HAS BEEN MADE FROM BEST AVAILABLE COPY.
- [ ] 2. A PORTION OF THE ORIGINAL DOCUMENT CONTAINS FINE DETAIL WHICH MAY MAKE READING OF PHOTOCOPY DIFFICULT.
- [ ] 3. THE ORIGINAL DOCUMENT CONTAINS COLOR, BUT DISTRIBUTION COPIES ARE AVAILABLE IN BLACK-AND-WHITE REPRODUCTION ONLY.
- [ ] 4. THE INITIAL DISTRIBUTION COPIES CONTAIN COLOR WHICH WILL BE SHOWN IN BLACK-AND-WHITE WHEN IT IS NECESSARY TO REPRINT.
- [ ] 5. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED, DOCUMENT WILL BE AVAILABLE IN MICROFICHE ONLY.
- [ ] 6. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED DOCUMENT WILL NOT BE AVAILABLE.
- [ ] 7. DOCUMENT IS AVAILABLE IN MICROFICHE ONLY.
- [ ] 8. DOCUMENT AVAILABLE ON LOAN FROM CFSTI (TT DOCUMENTS ONLY).
- [ ] 9.

NBS 9/64

PROCESSOR: [illegible]

AD 606935

INCREASING THE CAPACITY OF A NETWORK: THE PARAMETRIC BUDGET PROBLEM

D. R. Fulkerson

P-1401

12 June 1958

26-P

COPY 1 OF 1	
HARD COPY	$. 2.00
MICROFICHE	$. 0.50

The RAND Corporation
1700 MAIN ST. · SANTA MONICA · CALIFORNIA

Best
Available
Copy

SUMMARY

The problem considered in this paper is that of allocating a budget of resources among the links of a network for the purpose of increasing its flow capacity relative to given sources and sinks.

On the assumption that the cost of increasing each link capacity is linear, a labeling algorithm is described that permits rapid calculation of optimal allocations for all budgets.

INCREASING THE CAPACITY OF A NETWORK: THE PARAMETRIC BUDGET PROBLEM

1. Introduction. Suppose that a fixed budget can be allocated among the links of a network for the purpose of increasing its flow capacity relative to a given source and sink. How should the money be spent in order to maximize the resulting network capacity?

In this note we assume that the cost of increasing the capacity of a link is linear and homogeneous, which permits direct formulation of the problem described above as a linear program, and then describe an algorithm that produces solutions to the problem, not only for a fixed budget, but for all budgets, i.e., we solve the problem parametrically. The algorithm uses a variant of the labeling procedure previously developed to solve maximal network flow problems and minimal cost transportation problems [1–4].

It is interesting that, although the budget problem does not fall within the class of transportation-type programming problems, it can still be solved by a labeling procedure. Roughly speaking, the underlying reason for this is that, for a given budget problem, one can find a pair of transportation-type linear programs such that an optimal solution to the budget problem is given by a convex combination of certain optimal solutions to the two auxiliary problems. Indeed, our algorithm is designed to solve, efficiently, a sequence of such related transportation-type problems, the sequence having the property that adjacent pairs of solutions produced by the algorithm can be used to generate a solution of the parametric budget problem.

Section 2, below, contains a formulation of the budget problem as a linear program and a statement of the dual program. In Section 3 we set up the sequence of associated programs and include some heuristic discussion. Section 4 provides a statement of the algorithm. A numerical example illustrating the computation is given in Section 5. Section 6 concludes with proofs that the algorithm produces solutions to the associated programs, and to the budget problem.

2. The Budget Problem. We suppose given a network consisting of nodes P_0, P_1, ..., P_n and oriented links P_iP_j leading from P_i to P_j. Each link P_iP_j has associated with it two integers: c_{ij}, the existing flow capacity of the link, assumed nonnegative, and a_{ij}, the cost per unit of additional capacity, assumed positive. We take P_0 to be the source for flow, P_n the sink.[1]

Letting x_{ij} denote the flow from P_i to P_j along P_iP_j, y_{ij} the amount of capacity added to P_iP_j, b the total budget to be allocated for increased capacity, and v the net flow through the network from P_0 to P_n, the problem is to determine nonnegative values of x_{ij}, y_{ij}, v that

---

[1]We might equally well assume that there are several sources and sinks, provided we are interested in flows from any source to any sink. However, this situation can always be reduced to a single source and sink simply by joining all old sources to a new fictitious source by links of large capacity, and similarly for the sinks.

(1) maximize v

subject to the constraints

$$\text{(2a)}\quad \begin{cases} \sum_j (x_{0j} - x_{j0}) - v = 0 & \\ \sum_j (x_{ij} - x_{ji}) = 0 & (i = 1, \ldots, n-1) \\ \sum_j (x_{nj} - x_{jn}) + v = 0 & \end{cases}$$

$$\text{(2b)}\quad x_{ij} - y_{ij} \leq c_{ij}$$

$$\text{(2c)}\quad \sum_{i,j} a_{ij} y_{ij} = b .$$

Here, of course, b is assumed nonnegative.

Clearly this problem will not, in general, have integral solutions, because of the presence of constraint (2c). Nonetheless, almost all of the computation can be carried out in integers, as will be shown.

For future reference, we note that if we assign constraints (2a) the multipliers π_i $(i = 0, \ldots, n)$, constraints (2b) the multipliers γ_{ij}, and constraint (2c) the multiplier σ, one finds the dual of program (1) and (2) to be

$$\text{(3)}\quad \text{minimize} \quad \sum_{i,j} c_{ij} \gamma_{ij} + b\sigma$$

subject to

(4a) $$-\pi_0 + \pi_n \geq 1$$

(4b) $$\pi_i - \pi_j + \gamma_{ij} \geq 0$$

(4c) $$\sigma a_{ij} - \gamma_{ij} \geq 0$$

(4d) $$\gamma_{ij} \geq 0 .$$

If the nonnegative numbers x_{ij}, v satisfy equations (2a), we shall call x_{ij} a <u>flow</u> (from P_0 to P_n) and v the <u>flow value</u>.

3. <u>The Related Problems</u>. Consider the sequence of problems

(5) $$\text{maximize } tv - \sum_{i,j} a_{ij}y_{ij} \qquad (t = 1, 2, \ldots),$$

each subject to constraints (2a) and (2b) in nonnegative variables.

Notice that for t sufficiently large, e.g., if t is greater than the cost of adding a unit of capacity to each link of a chain from P_0 to P_n, the form (5) is unbounded on the convex set defined by (2a) and (2b). Thus the sequence of related problems we will need to consider is finite. We let T denote the largest value of t for which the form (5) is bounded.

Now suppose x_{ij}^t, y_{ij}^t, v^t solve the t-th one of these problems, and define

$$b^t = \sum_{i,j} a_{ij} y_{ij}^t , \qquad t = 1, \ldots, T .$$

Then it is easy to see that x_{ij}^t, y_{ij}^t, v^t solve the budget problem for $b = b^t$. Moreover, the numbers b^t will be monotone non-decreasing in t. It might therefore seem plausible that if we are given b such that $b^t \leq b \leq b^{t+1}$, then a solution to such an intermediate budget problem could be generated by expressing b as a convex combination of b^t and b^{t+1}, and taking the same convex combination of the solutions x_{ij}^t, y_{ij}^t, v^t and x_{ij}^{t+1}, y_{ij}^{t+1}, v^{t+1} . This turns out to be almost right — that is, it is false that any two such solutions can be used in this way to solve an intermediate budget problem, but it is true that there exist solutions to the t-th and (t+1)-th related problems that do generate solutions for all b lying in the interval (b^t, b^{t+1}) associated with these particular solutions.

The algorithm of the next section will, in fact, be shown to produce integral solutions x_{ij}^t, y_{ij}^t, v^t $(t = 1, \ldots, T)$ and hence a set of integers $0 = b^1 \leq b^2 \leq \ldots \leq b^T$, such that

(a) if $b^t \leq b \leq b^{t+1}$, then a solution to the budget problem corresponding to b is given by a convex combination of x_{ij}^t, y_{ij}^t, v^t and x_{ij}^{t+1}, y_{ij}^{t+1}, v^{t+1};

(b) if $b > b^T$, a solution can be obtained from x_{ij}^T, y_{ij}^T, v^T.

Moreover, the computation for the related problem t begins with the solution previously generated for problem $t-1$, and thus the entire set of "spanning" solutions for the budget problems can be obtained efficiently.

4. <u>The Algorithm</u>. Before stating the algorithm for solving the sequence of related problems, we note that the dual of problem t is to find numbers π_i^t, one for each node P_i, and γ_{ij}^t, one for each arc P_iP_j, that

(6) $$\text{minimize} \sum_{i,j} c_{ij}\, \gamma_{ij}^t$$

subject to the constraints

(7a) $$-\pi_0^t + \pi_n^t \geq t$$

(7b) $$\pi_i^t - \pi_j^t + \gamma_{ij}^t \geq 0$$

(7c) $$0 \leq \gamma_{ij}^t \leq a_{ij} .$$

It follows that feasible solutions x_{ij}^t, y_{ij}^t, v^t and π_i^t, γ_{ij}^t to the primal and dual problems, respectively, which satisfy the conditions

(8a) $$\pi_0^t = 0, \ \pi_n^t = t$$

(8b) $$\pi_i^t - \pi_j^t + \gamma_{ij}^t > 0 \Longrightarrow x_{ij}^t = 0$$

(8c) $$\gamma_{ij}^t > 0 \implies x_{ij}^t - y_{ij}^t = c_{ij}$$

(8d) $$\gamma_{ij}^t < a_{ij} \implies y_{ij}^t = 0,$$

are optimal solutions.

The dual variables γ_{ij}^t and primal variables y_{ij}^t need not be mentioned explicitly in describing the computation. Instead, we shall deal only with node numbers π_i^t and flows x_{ij}^t, and will construct these to satisfy

(9a) $$\pi_0^t = 0, \ \pi_n^t = t$$

(9b) $$\pi_j^t - \pi_i^t \leq a_{ij}$$

(9c) $$\pi_j^t - \pi_i^t = 0 \implies x_{ij}^t \leq c_{ij}$$

(9d) $$\pi_j^t - \pi_i^t = a_{ij} \implies x_{ij}^t \geq c_{ij}$$

(9e) $$\pi_j^t - \pi_i^t < 0 \implies x_{ij}^t = 0$$

(9f) $$0 < \pi_j^t - \pi_i^t < a_{ij} \implies x_{ij}^t = c_{ij} .$$

In addition, all variables will have integral values.

It is easy to check that if there are node numbers π_i^t and a flow x_{ij}^t such that (9a) – (9f) hold, then by defining

(10) $$\gamma_{ij}^{t} = \max \left(0, \pi_j^t - \pi_i^t\right)$$

(11) $$y_{ij}^{t} = \max \left(0, x_{ij}^t - c_{ij}\right),$$

one has feasible solutions to both primal and dual problems that satisfy (8a)–(8d), and hence are optimal.

To start the computation, take $\pi_i^0 = 0$ and $x_{ij}^0 = 0$. These clearly satisfy conditions (9) for $t = 0$. The computation now progresses by a sequence of "labelings" (Step A below), each of which can terminate in one of three ways: "finite breakthrough," in which case the flow is changed (Step B), "nonbreakthrough," in which case the node integers are changed (Step C), or "infinite breakthrough," in which case the computation ends, and T has been discovered.

The inputs for the t-th application of the routine composed of Steps A, B, C are π_i^{t-1}, x_{ij}^{t-1}. The node numbers π_i^{t-1} are used to divide the links P_iP_j of the network into three classes as follows. A link P_iP_j is <u>0-admissible</u>, <u>a-admissible</u>, or <u>inadmissible</u> according as the value of $\pi_j^{t-1} - \pi_i^{t-1}$ is 0, a_{ij}, or neither of these.[2]

<u>Step A</u>. (Labeling process).

(1) Assign P_0 the label (P_{n+1}^{+}, ∞); consider P_0 as unscanned.

---

[2]Thus initially all links are 0-admissible. Steps A, B, C, then reduce to the algorithm of ref. [1] for constructing a flow of maximal value in a network with capacity limitations c_{ij} on links.

(2) Take any labeled, unscanned node P_i; suppose it is labeled (P_k^+, ∞). (Initially P_0 will be the only such.) To all nodes P_j that are unlabeled and such that P_iP_j is a - admissible, assign the label (P_i^+, ∞). Consider P_i as scanned and the newly labeled P_j, if any, as unscanned. Repeat until either the sink P_n has been labeled (infinite breakthrough), or until no new labels are possible and this is not the case. In the former case, terminate; in the latter case, proceed to (3) below.

(3) (At this stage we have a labeled set of nodes including P_0 but not P_n, and each has a label of the form (P_k^+, ∞).) All nodes now revert to the unscanned state, and the labeling process continues as follows. Take any labeled, unscanned node P_i; suppose it is labeled $(P_k^{\pm}, h)$. (Initially we have only labels of the form (P_k^+, ∞).) To all nodes P_j that are unlabeled, such that P_iP_j is 0 - admissible, and $x_{ij}^{t-1} < c_{ij}$, assign the label $(P_i^+, \min(h, c_{ij} - x_{ij}^{t-1}))$. To all nodes P_j that are now unlabeled, such that P_jP_i is 0 - admissible, and $x_{ji}^{t-1} > 0$, assign the label $(P_i^-, \min(h, x_{ji}^{t-1}))$. Next, if P_j is unlabeled and P_iP_j is a - admissible, label P_j with (P_i^+, h). (Initially, when we are labeling from a node of the starting set, this case cannot occur.) Finally, if P_j is unlabeled, P_jP_i is a - admissible, and $x_{ji}^{t-1} > c_{ji}$, label P_j with $(P_i^-, \min(h, x_{ji}^{t-1} - c_{ji}))$. Consider P_i as scanned and the newly labeled P_j, if any, as unscanned. Repeat until either

the sink P_n has been labeled with, say, (P_k^+, h)[3], or until no new labels are possible and this is not the case. In the former case (finite breakthrough), go to Step B. In the latter case (nonbreakthrough), go to Step C.

Step B. (Flow change).

(Here the sink P_n has been labeled with (P_k^+, h).) Replace x_{kn}^{t-1} by $x_{kn}^{t-1} + h$, and go on to P_k and its label. In general, if P_k is labeled (P_j^+, ℓ), replace x_{jk}^{t-1} by $x_{jk}^{t-1} + h$, and if labeled (P_j^-, ℓ), replace x_{kj}^{t-1} by $x_{kj}^{t-1} - h$, in either case turning attention then to P_j and its label. Stop the flow change when P_0 has been reached. Now discard the labels generated in (3) of Step A and repeat A3 with the new flow in place of x_{ij}^{t-1}.

Step C. (Node number change).

(The labeling process has resulted in nonbreakthrough.) Give the present flow (which may or may not be x_{ij}^{t-1}) the name x_{ij}^t and define node numbers π_i^t by

$$\pi_i^t = \begin{cases} \pi_i^{t-1} & \text{if } P_i \text{ is labeled} \\ \pi_i^{t-1} + 1 & \text{if } P_i \text{ is unlabeled.} \end{cases}$$

The entire routine is then repeated using π_i^t and x_{ij}^t as inputs.

---

[3]The sink P_n will never receive a label of the form (P_k^-, h), since every flow generated by the algorithm will have $x_{nj} = 0$. Similarly each flow will have $x_{j0} = 0$, so that any node P_j labeled from P_0 will have a label of the form (P_0^+, h).

In the concluding section we shall sketch proofs that the flows x^t_{ij} generated in the computation have the properties discussed in Section 3, but perhaps some preliminary explanatory comments are in order.

The labeling process A1-A2 is a search for a chain from P_0 to P_n of a-admissible links. If none such exists, we proceed to enlarge the search (A3) in an attempt to find a path from P_0 to P_n of admissible links (where the word "path", as opposed to "chain", means that a link may be traversed opposite its orientation in going from P_0 to P_n) having the property that the (integral) flow change h made along the path (Step B) is positive and yields a flow again satisfying (9c) and (9d). Inadmissible links correspond to (9e) and (9f), and in these we keep the flow fixed, so that these conditions are also maintained. Thus if we enter the routine with node numbers π^{t-1}_i and a flow x^{t-1}_{ij} satisfying (9c)-(9f), the same node numbers π^{t-1}_i and the output flow x^t_{ij} still satisfy (9c)-(9f), and consequently the output flow will again be a solution to related problem $t-1$. In addition, it is a solution to related problem t (as can be shown using the transformation of node numbers given in Step C), and hence we can repeat the process. It is this fact — that x^t_{ij} solves both problems $t-1$ and t — which enables one to prove that the sequence of flows $x^1_{ij}, \ldots, x^T_{ij}$ produced by the algorithm are spanning solutions for the budget problem.

5. An Example. Let the network be that of Fig. 1 below, the capacity c_{ij} of link P_iP_j being the number in the upper left of the box, and the cost a_{ij} of adding one unit of capacity being the number in the upper right.[4] Assume that we have

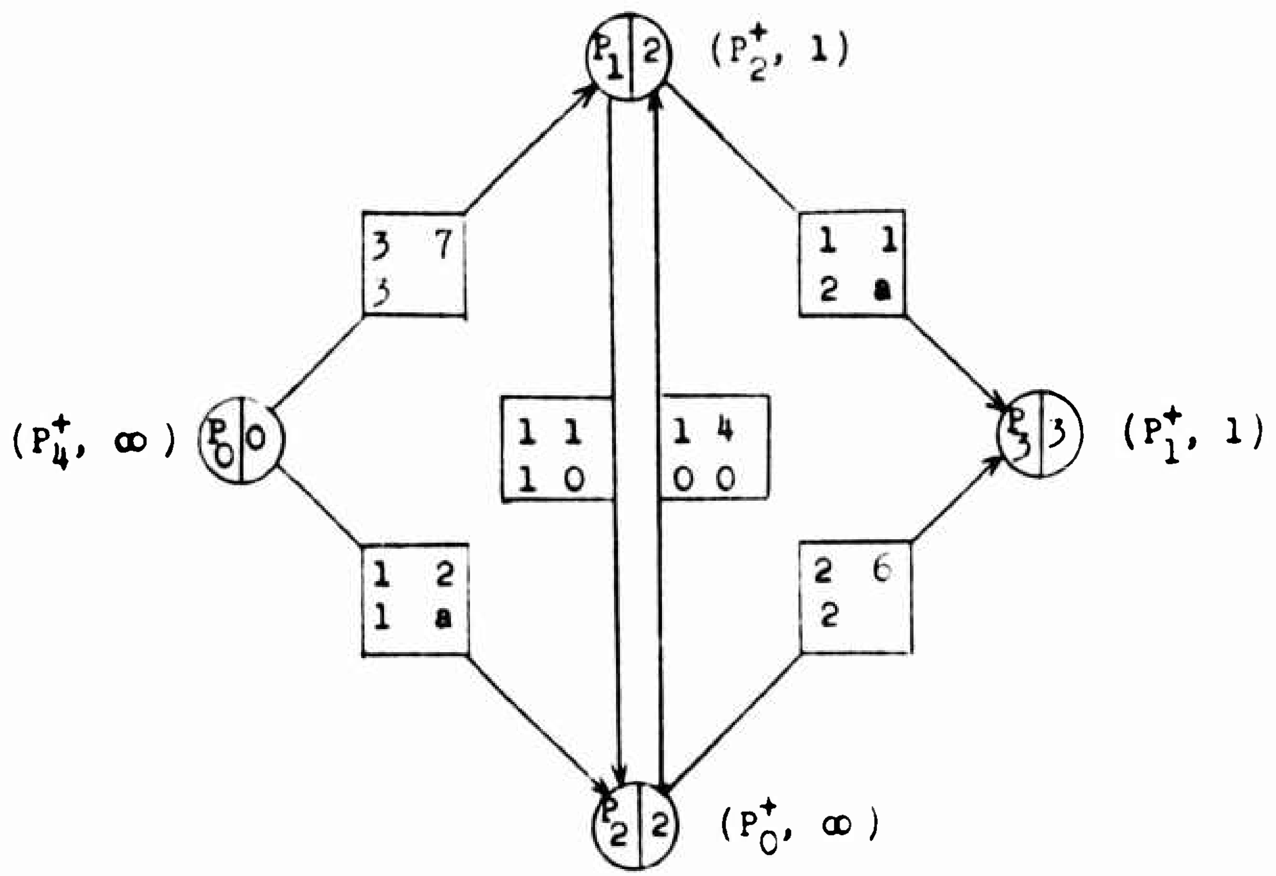

Fig. 1

the node numbers π_i^3 shown in the figure, and the flow x_{ij}^3 indicated by the numbers in the lower left of the boxes, and wish to compute x_{ij}^4 and π_i^4. Using the numbers π_i^3, we divide the links into the three classes: 0-admissible (indicated in the figure by a zero in the lower right of the box), a-admissible (indicated by an a in the lower right of the box), and inadmissible (indicated by no entry in the lower right of the box).

[4] Links not shown in Fig. 1 are assumed to have zero capacity and large cost for additional capacity.

The labeling process A1—A2 yields the labels (P_4^+, ∞) on P_0 and (P_0^+, ∞) on P_2. We then go on to A3. Scanning P_0 gives no more labels, but from P_2 we can label P_1 with $(P_2^+, \min(\infty, 1))$, and this completes the scanning of P_2. (Notice that P_1 could also have been labeled with $(P_2^-, \min(\infty, 1))$, since the order in which the labeling rules of A3 are applied is immaterial.) Finally, from P_1 we break through to P_3 with the label $(P_1^+, 1)$, and have thus located a chain, found by tracing the labels backward from P_3, along which we can increase the flow by an additional unit.

After changing the flow, discarding the old labels, and relabeling, we obtain the labels shown in Fig. 2 below. Again

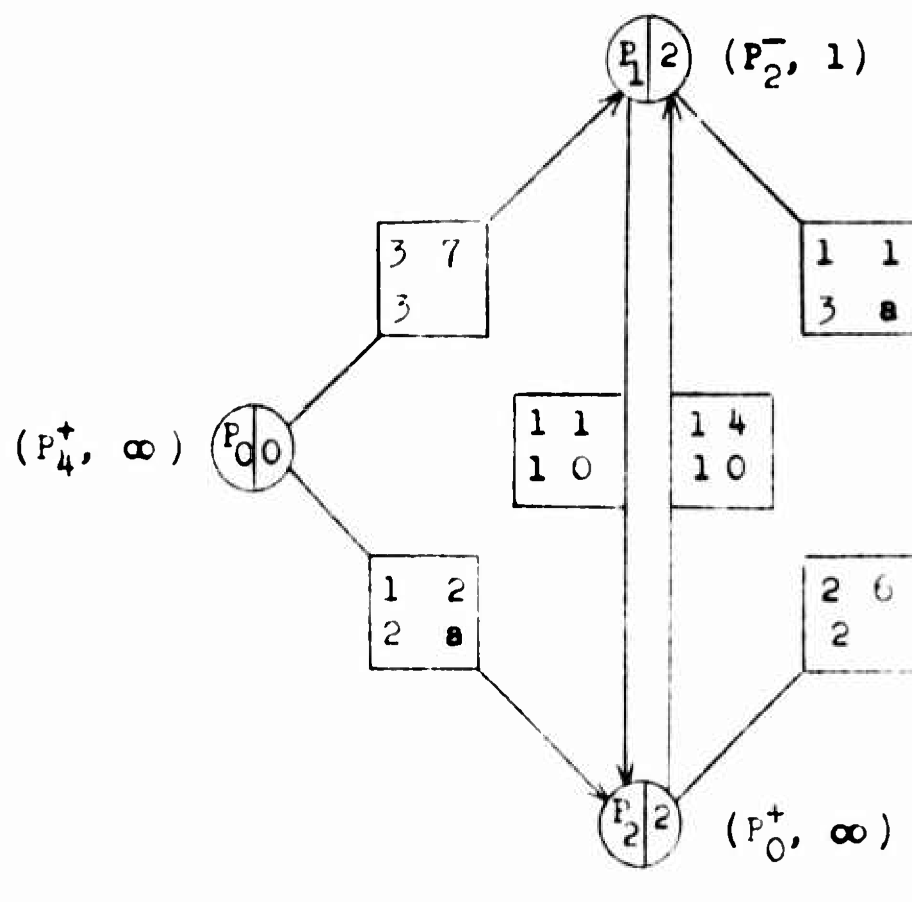

Fig. 2

we have a finite breakthrough, and therefore change the flow along the path indicated by the labels: add 1 to x_{13}, subtract 1 from x_{12}, and add 1 to x_{02}. We then relabel, obtaining the labels shown in Fig. 3 below. This time we have a non-breakthrough, and thus go to Step C, the node-number change.

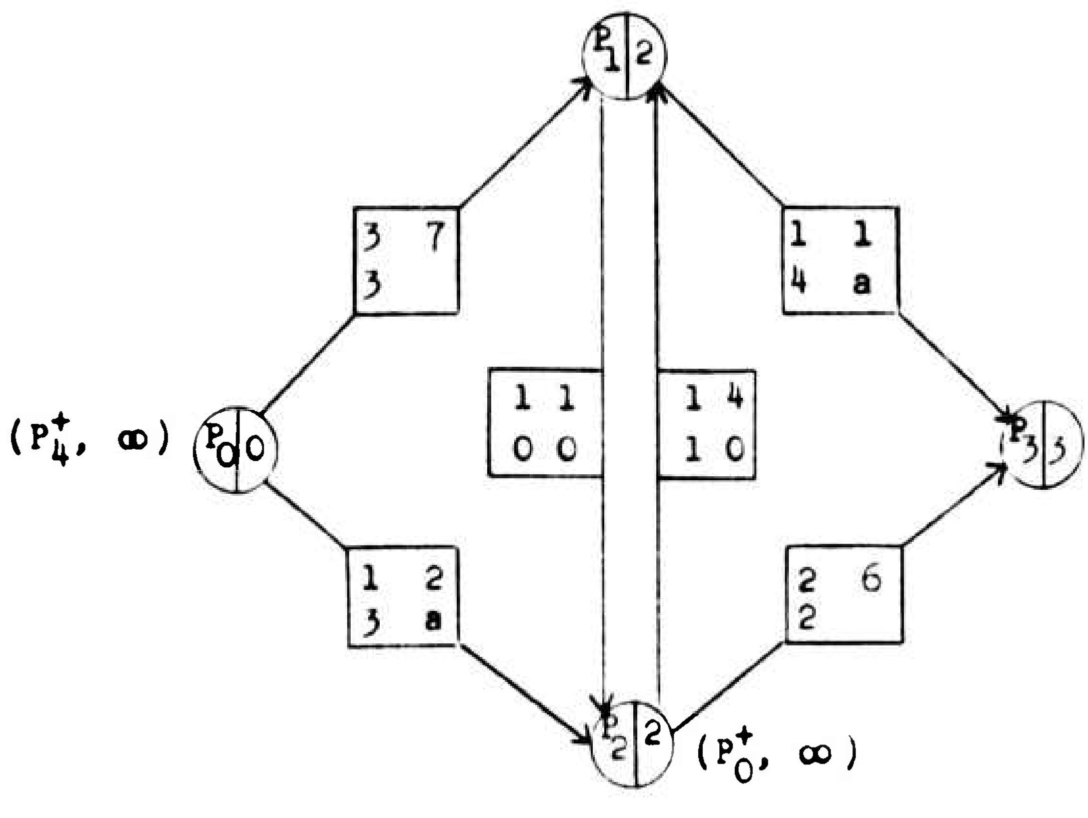

Fig. 3

The flow shown in Fig. 3 is therefore x_{ij}^4, and the new node numbers π_i^4 are given by adding 1 to the numbers on unlabeled nodes P_1 and P_3: $\pi_0^4 = 0$, $\pi_1^4 = 3$, $\pi_2^4 = 2$, $\pi_3^4 = 4$.

Observe that

$$\sum_{i,j} a_{ij} y_{ij}^4 = 2 y_{02}^4 + 1 y_{13}^4 = 7,$$

and thus if we are given a budget $b = 7$, we should boost the capacity of P_0P_2 by 2 units, that of P_1P_3 by 3 units, thereby achieving a total flow of 6 units from P_0 to P_3. On the other hand, we see from Fig. 1 that

$$\sum_{i,j} a_{ij} y_{ij}^3 = 1 y_{13}^3 = 1,$$

so that with $b = 1$, the capacity of P_1P_3 should be increased by 1 unit, permitting a total flow of 4 units through the network. Notice also that

$$3v^3 - \sum_{i,j} a_{ij} y_{ij}^3 = 11 = 3v^4 - \sum_{i,j} a_{ij} y_{ij}^4$$

and hence x_{ij}^4 solves related problem 3 provided x_{ij}^3 does.

6. Theorems and Proofs. It is not difficult to see that if we enter Step A with a flow x_{ij} and obtain new numbers x'_{ij} via Step B, then x'_{ij} is a flow also, since it is obtained from x_{ij} by adding a positive amount h to the flow in links of a path from P_0 to P_n that are traversed with their orientation (in going from P_0 to P_n), and subtracting h from the flows in links traversed against their orientation. Moreover, h is no greater than the minimum of the link flows in the reverse oriented links of the path, so that nonnegativity is maintained.

The routine composed of Steps A, B, C terminates. For if A1 and A2 do not locate a chain of a-admissible links from P_0 to P_n, let L be the set of indices of nodes that are labeled in A1 and A2. Thus $0 \in L$, $n \notin L$. Now any flow x_{ij} produced via A3 and B satisfies $x_{ij} \leq c_{ij}$ for all links P_iP_j that are not a-admissible. Hence, summing equations (2a) over $i \in L$ yields

$$v = \sum_{\substack{i \in L \\ j \notin L}} (x_{ij} - x_{ji}) \leq \sum_{\substack{i \in L \\ j \notin L}} x_{ij}$$

and thus, since links P_iP_j for $i \in L$, $j \notin L$ are not a-admissible, we have

$$v \leq \sum_{\substack{i \in L \\ j \notin L}} c_{ij} .$$

Consequently, since v increases by $h \geq 1$ with each occurrence of a flow change, there can be only finitely many of these.

Thus, starting with the flow $x_{ij}^0 = 0$, the algorithm successively produces flows x_{ij}^t for $t > 0$.

<u>Theorem 1. The flows</u> x_{ij}^t <u>produced by the algorithm and the corresponding</u> $y_{ij}^t = \max(0, x_{ij}^t - c_{ij})$, $v^t = \sum_j(x_{0j}^t - x_{j0}^t)$, <u>maximize the form</u> $tv - \sum_{i,j} a_{ij}\, y_{ij}$ <u>subject to constraints</u> (2a), (2b) <u>in nonnegative variables</u>, i.e. x_{ij}^t, y_{ij}^t, <u>and</u> v^t <u>solve related problem</u> t .

It suffices to show that π_i^t, x_{ij}^t satisfy (9a) – (9f) . Since it is clear that

$$\pi_i^0 = 0,\ x_{ij}^0 = 0,\ y_{ij}^0 = 0,\ v^0 = 0$$

satisfy (9a) – (9f) with $t = 0$, we may proceed by induction on t .

Property (9a) is clear from the induction assumption $\pi_0^{t-1} = 0$, $\pi_n^{t-1} = t - 1$, the node number change of Step C, and the fact that P_0 is labeled and P_n unlabeled in case of nonbreakthrough.

Consider (9b). Since $\pi_j^{t-1} - \pi_i^{t-1} \leq a_{ij}$, then $\pi_j^t - \pi_i^t$ could exceed a_{ij} only if $\pi_j^{t-1} - \pi_i^{t-1} = a_{ij}$ and $\pi_j^t = \pi_j^{t-1} + 1$, $\pi_i^t = \pi_i^{t-1}$. But then P_iP_j is a-admissible, P_i is labeled and P_j unlabeled at the conclusion of labeling, a contradiction.

For (9c), suppose $\pi_j^t - \pi_i^t = 0$, and consider cases. If $\pi_j^{t-1} - \pi_i^{t-1} < 0$, so that $x_{ij}^{t-1} = 0$, then, since P_iP_j is inadmissible, we also have $x_{ij}^t = x_{ij}^{t-1} = 0 \leq c_{ij}$. If $\pi_j^{t-1} - \pi_i^{t-1} = 0$, so that $x_{ij}^{t-1} \leq c_{ij}$, again we have $x_{ij}^t \leq c_{ij}$, since x_{ij}^{t-1} can be increased by at most $c_{ij} - x_{ij}^{t-1}$ in a sequence of flow changes. If $0 < \pi_j^{t-1} - \pi_i^{t-1} < a_{ij}$, then P_iP_j is inadmissible and consequently $x_{ij}^t = x_{ij}^{t-1} = c_{ij}$. Finally, if $\pi_j^{t-1} - \pi_i^{t-1} = a_{ij}$, then $\pi_i^t = \pi_i^{t-1} + 1$, $\pi_j^t = \pi_j^{t-1}$, and hence P_i is unlabeled, P_j labeled at the conclusion of labeling. But if $x_{ij}^t > c_{ij}$, this is a contradiction, since P_iP_j is a-admissible. Hence $x_{ij}^t \leq c_{ij}$. This completes the proof of (9c).

Proofs of the remaining properties can be given along similar lines, and so we omit them.

<u>Corollary</u>. <u>The flow</u> x_{ij}^t <u>and its corresponding</u> y_{ij}^t, v^t <u>solve related problem</u> $t - 1$.

This follows from the fact that x_{ij}^{t-1}, y_{ij}^{t-1}, v^{t-1} solve related problem $t - 1$ and the remarks at the end of Sec. 4.

Suppose that the algorithm terminates after the Tth application of the routine composed of steps A, B, C, i.e. we enter step A with π_i^T, x_{ij}^T and infinite breakthrough occurs. Thus a chain of a-admissible links from P_0 to P_n, say

$$(12) \qquad P_{i_0}P_{i_1},\ P_{i_1}P_{i_2}, \ldots,\ P_{i_{k-1}}P_{i_k} \qquad (i_0 = 0,\ i_k = n) ,$$

has been located, and hence from (9a) and the definition of a-admissibility, it follows that

$$(13) \qquad T = \pi_n^T - \pi_0^T = \sum_{\ell=0}^{k} \left(\pi_{i_{\ell+1}}^T - \pi_{i_\ell}^T\right) = \sum_{\ell=0}^{k} a_{i_\ell i_{\ell+1}} .$$

Consequently this chain, of "a-length" T, has minimal a-length over all chains from P_0 to P_n, since if T were greater than the a-length of some chain, the form $Tv - \sum_{i,j} a_{ij} y_{ij}$ would obviously be unbounded, contradicting the maximality of $Tv^T - \sum_{ij} a_{ij} y_{ij}^T$.

Let $b^t = \sum_{i,j} a_{ij} y_{ij}^t$ $(t = 1, \ldots, T)$ be the successive values of $\Sigma a_{ij} y_{ij}$ produced by the algorithm. Then

$$0 = b^1 \leq b^2 \leq \ldots \leq b^T .$$

For on the first application of the algorithm, all links are O-admissible, hence $x_{ij}^1 \leq c_{ij}$, or $y_{ij}^1 = 0$. To establish the monotoneity, assume that $b^t < b^{t-1}$. Since y_{ij}^{t-1}, v^{t-1} and y_{ij}^t, v^t are respectively maximal in problems $t-1$ and t, we have

$$(t-1)\ v^{t-1} - b^{t-1} \geq (t-1)\ v^t - b^t$$

$$t\ v^t - b^t \geq t\ v^{t-1} - b^{t-1},$$

whence adding gives

$$v^{t-1} \leq v^t ,$$

an inequality that is also clear directly from the algorithm. Thus, if $b^t < b^{t-1}$, we get

$$(t-1)\ v^t - b^t > (t-1)\ v^{t-1} - b^{t-1},$$

a contradiction.

__Theorem 2__. __Let__ $b = ab^t + (1-a)\ b^{t+1}$, $0 \leq a \leq 1$. __Then__

$$x_{ij} = ax_{ij}^t + (1-a)\ x_{ij}^{t+1}$$

$$y_{ij} = ay_{ij}^t + (1-a)\ y_{ij}^{t+1}$$

$$v = av^t + (1-a)\ v^{t+1}$$

__solve__ (1) __and__ (2). __If, on the other hand, we have__ $b > b^T$, __then the flow__ x'_{ij} __and its corresponding__ y'_{ij}, v' __obtained from__

x_{ij}^T, y_{ij}^T, v^T by adding $\frac{1}{T}$ $(b - b^T)$ units of flow along the a-admissible chain (12), solve (1) and (2).

While Theorem 2 can be proved directly, we choose to give a proof using the dual problem (3) and (4) in order to point out how to obtain solutions to the dual of the budget problem from the node numbers generated in the algorithm.

Inasmuch as π_i^t and the associated γ_{ij}^t given by (10) satisfy the constraints (7), it follows that

$$(14) \qquad \pi_i = \frac{\pi_i^t}{t}, \quad \gamma_{ij} = \frac{\gamma_{ij}^t}{t}, \quad \sigma = \frac{1}{t}$$

satisfy the constraints (4). Moreover, we have

$$\sum_{i,j} c_{ij}\, \gamma_{ij}^t = t\, v^t - b^t,$$

since π_i^t, γ_{ij}^t are optimal for (6) and (7). Thus

$$\begin{aligned}\sum_{i,j} c_{ij}\, \gamma_{ij} + b\sigma &= \frac{1}{t}\left(\sum_{i,j} c_{ij}\, \gamma_{ij}^t + b\right) \\ &= \frac{1}{t}\,(t\, v^t - b^t + b) \\ &= v^t + \frac{1}{t}\,(b - b^t)\ .\end{aligned}$$

Now since x_{ij}^{t+1}, y_{ij}^{t+1}, v^{t+1} and x_{ij}^t, y_{ij}^t, v^t both solve problem t, we have

$$t\ v^{t+1} - b^{t+1} = t\ v^{t} - b^{t} .$$

Thus if $b^{t+1} = b^{t} = b$, then $v^{t+1} = v^{t} = v$, and hence $\Sigma\, c_{ij}\, \gamma_{ij} + b\sigma = v$. If, on the other hand, $b^{t} < b^{t+1}$, we have

$$\frac{1}{t} = \frac{v^{t+1} - v^{t}}{b^{t+1} - b^{t}} ,$$

so that

$$\begin{aligned} v &= v^{t} + (1 - \alpha)\ (v^{t+1} - v^{t}) \\ &= v^{t} + \frac{(b - b^{t})}{b^{t+1} - b^{t}}\ (v^{t+1} - v^{t}) \\ &= v^{t} + \frac{1}{t}\ (b - b^{t}) . \end{aligned}$$

Thus in either case, we see that

$$\sum_{i,j} c_{ij}\, \gamma_{ij} + b\sigma = v . \tag{15}$$

Hence, since x_{ij}, y_{ij}, v satisfy (2), and π_i, γ_{ij}, σ satisfy (4), it follows from (15) that they constitute optimal dual solutions.

Suppose, finally, that $b > b^{T}$. It follows from (9d) and the existence of the a-admissible chain (12) that

$$\Sigma\, a_{ij}\, y'_{ij} = b^T + \frac{1}{T}\,(b - b^T) \sum_{\ell=0}^{k} a_{i_\ell i_{\ell+1}} ,$$

and hence from (13) we have

$$\Sigma\, a_{ij}\, y'_{ij} = b\ .$$

Thus x'_{ij} , y'_{ij} , v' satisfy (2) . Defining

$$\pi'_i = \frac{\pi_i^T}{T},\quad \gamma'_{ij} = \frac{\gamma_{ij}^T}{T},\quad \sigma' = \frac{1}{T} \tag{16}$$

again gives a pair of optimal dual solutions to the budget problem.

REFERENCES

1. Ford, L. R., Jr., and D. R. Fulkerson, "A simple algorithm for finding maximal network flows and an application to the Hitchcock problem," Can. J. Math., Vol. 9, 1957, pp. 210–218.

2. ————————, "Solving the transportation problem, Man. Sci., Vol. 3, No. 1, 1956, pp. 24–32.

3. ————————, "A primal dual algorithm for the capacitated Hitchcock problem," N. R. L. Q., Vol. 4, No. 1, 1957, pp. 47–54.

4. ————————, "Constructing maximal dynamic flows from static flows," to appear in J. O. R. S. A., June, 1958.